REMARKS

The acknowledgment of the claim for foreign priority under 35 U.S.C. §119 and receipt of the priority document is noted with appreciation.

The specification was objected to by the Examiner on the ground that the "BRIEF SUMMARY OF THE INVENTION" on pages 4–6 does not meet the objectives of the Manual of Patent Examining Procedure (MPEP), citing MPEP 608.01 and 1302.01 and 37 C.F.R. §§1.73 and 1.75. The page numbers are pages 5–7, not 4–6 as stated by the Examiner. In response, the paragraphs beginning on page 5, line 8, to page 7, line 4 have been amended. The "BRIEF SUMMARY OF THE INVENTION", as amended, is believed to meet the objectives of the MPEP and C.F.R. and, therefore, withdrawal of the objection to the specification is respectfully requested.

The disclosed and claimed invention solves a specific and important problem in the design and manufacture of large scale integrated (LSI) circuits, such as SOC (System On a Chip). These are very complex circuits, typically comprising tens of millions of transistors arranged in many circuit modules to form a complete system. An example of such a system is illustrated in Figure 8 comprising two CPUs (Central Processing Units), a timer and a counter connected by a PCI (Peripheral Component Interconnect) bus. To this system might be added one or more memory modules, and all of this fabricated on a single chip. The design and manufacture of such systems is very complex, time consuming and expensive. It is therefore important to determine that a design is correct before committing it to the manufacturing process. This is done by simulation, and conventional software simulators use HDL (Hardware Description Language) for simulating a system design. While HDL has proved to be quite useful in designing and testing small scale circuits or systems, its application to LSI circuits has several drawbacks enumerated on page 3 of the specification.

The disclosed and claimed invention uses an object oriented programming language for simulating a system by easily and reliably describing the circuit

modules and their connections. As background, it should first be understood what characterizes an object oriented language. For this purpose, there is attached copies of pages 38 to 70 of the book *Object Oriented Design* by Grady Booch, published by The Benjamin/Cummings Publishing Company, Inc. (1991). As set out on page 38, there are four major components to an object oriented language: abstraction, encapsulation, modularity, and hierarchy. There are also three minor elements: typing, concurrency and persistence. These are each described in some further detail on the succeeding pages. Also attached is page 473 from the Grady Booch book which provides a diagram of the genealogy of object-based and object-oriented programming languages.

As noted on page 4 of the specification, there have been prior attempts to use a language having features of an object oriented language or an object oriented language to simulate LSI circuits. However, these prior attempts have not made use of the advantages of the object oriented languages for this type of application.

Claims 1 to 24 now appear in the application. By this amendment, claim 3 has been amended and new claims 11 to 24 have been added. Claims 9 and 10 were previously amended. New claims 11 to 17 correspond to original claims 2 to 8 but are dependent on claim 9, while new claims 18 to 24 correspond to original claims 2 to 8 but are dependent on claim 10.

Claims 1 to 8 and 10 were rejected under 35 U.S.C. §112, second paragraph, as being incomplete for omitting essential steps. This rejection is respectfully traversed.

As to claim 1, the Examiner states that the recitation of "inheriting the circuit base classes prepared by the library" is passive and implies that one or more actions must be initiated elsewhere to cause the circuit base classes to be inherited or received and that the steps, which initiate these actions, are missing. The Examiner is referred to the attached pages from the book by Grady Booch entitled *Object Oriented Design with Applications*, The Benjamin/Cummings Publishing Company, Inc. (1991), and in particular, to pages 54 to 59, which deal with the meaning of hierarchy. Beginning at the bottom of page 54, it is stated that

> "Inheritance is the most important 'kind of' hierarchy, and . . . it is an essential element of object-oriented systems. Basically, inheritance defines a relationship among classes, wherein one class shares the structure or behavior defined in one or more classes (called *single inheritance* and *multiple inheritance*, respectively). Inheritance thus represents a hierarchy of abstractions, in which a subclass inherits from one or more superclasses. Typically, a subclass augments or redefines the existing structure and behavior of its superclasses."

In the case of the present invention, the process is illustrated in Figures 2 and 7. With reference to Figure 7, it will be observed that the double arrow indicates inheritance. The term "inheriting", as used in the art of object oriented languages, is not passive, as stated by the Examiner, but rather has a well understood meaning which does not imply any missing steps or actions, as clearly demonstrated by the Grady Booch book.

The same comments apply to claim 10.

Claim 9 was rejected under 35 U.S.C. §112, second paragraph, as being incomplete for omitting essential elements. This rejection is also respectfully traversed for the reasons advanced above.

Claims 1 to 10 were rejected under 35 U.S.C. §103(a) as bing unpatentable over U.S. Patent No. 4,965,743 to Malin. This rejection is respectfully traversed for the reason that Malin does not show, suggest or otherwise teach the claimed invention.

What Malin discloses is a simulation tool implementing an element of artificial intelligence to create computer models simulating continuous activities, functions and/or behavior, especially for supporting analysis of a range of engineered space systems, such as a two-phase thermal bus system or an electrochemical air purification system, as typically involved in NASA's space programs. To that end, the preferred embodiment employs an object-oriented language (identified as KEE™ by IntelliCorp) and a discrete event simulator

(identified as SimKit™ by IntelliCorp) with additional functionality provided by the underlying LISP programming environment (identified as provided by Symbolics). It is noted here in passing that LISP is a well known artificial intelligence (AI) language but is not itself an object oriented language (reference is made to page 473 of the attached pages from the Grady Booch book). It is assumed from this description in column 2 of Malin et al. that "KEE" is an object oriented language, but no mention of it can be found of this programming language in the literature nor does the Malin et al. patent provide a description of the language. Furthermore, IntelliCorp's Web site makes no reference to it. Attached are copies of screenprints from IntelliCorp's Web site, including their home page and products page. In addition, Intellicorp's Web site makes no reference to "SimKit", but ROSIM Systems of the United Kingdom does, suggesting that perhaps ROSIM Systems may have purchased the product from IntelliCorp, if the reference in the Malin et al. patent is correct. Attached are copies of screenprints from ROSIM Systems' Web site. There, SimKit is described as a software based simulation tool of the type generally described in the Malin et al. patent. In any case, what is presented in Malin et al. is (1) an unidentifiable object oriented language, (2) a commercially available software simulation package for general purpose discrete event simulation, and (3) an AI software language. There is no explanation of how the unidentifiable object oriented language is used or how the three components are integrated to perform the simulation function.

More importantly, Malin et al. do not address the specific problem solved by the disclosed and claimed invention, which to provide a method and apparatus for simulating a system having a plurality of circuit modules using software. While Malin et al. purport to provide a general purpose modeling tool, they describe modeling of systems employing condensers, evaporators, filters, pipes, pumps, regenerators, subcoolers, and valves. There is no hint of simulating LSI circuits which involves a much different set of considerations. The Examiner has focused on a number of words in Malin et al. which are common to this disclosure

and to object oriented languages in general, but it is quite unclear from Malin et al. whether objected oriented language or artificial intelligence (AI), i.e., LISP, are involved, the latter not being an object oriented language. The Examiner will recognize, for example, that "library" has different meaning in terms of content between the two. In the Malin et al. patent, the "library design module supports the building of library *knowledge* including component classes and elements pertinent to a particular domain of continuous activities, functions and behavior being modelled [sic]" (Abstract, emphasis added). This suggests a knowledge base for an AI system, not a library in the meaning and sense of an object oriented language.

In contrast to the Malin et al. patent, Applicants have described the preferred embodiment of their invention in terms of a well known and well documented object oriented language, C++. Moreover, Applicants have provided specific, detailed examples of the use of the this language for the specific purpose for which the claimed invention was developed; that is, the simulation of an LSI system. The claims are specific to this problem, a problem not even recognized by Malin et al. Thus, claim 1 is directed to a "method for simulating a system having a plurality of circuit modules using software". This method involves "using an object oriented language" and "preparing a plurality of circuit base classes, which describe base circuit modules as classes, as a library". Once the library has been prepared, then "the circuit modules to be simulated [are described] as classes by *inheriting* the circuit base classes prepared as the library" (emphasis added), and "the system to be simulated [is described] by combining the circuit modules described as the classes". Nothing like this is remotely suggested by Malin et al. Independent claims 9 and 10 contain similar limitations and define over the Malin et al. patent for the same reasons.

The prior art made of record and not relied upon has been considered; however, only the patent to Balakrishnan et al. appears to be even remotely relevant to the disclosed and claimed invention. Balakrishnan et al. described a variant of the prior art HDL (Hardware Design Language), which is not an object

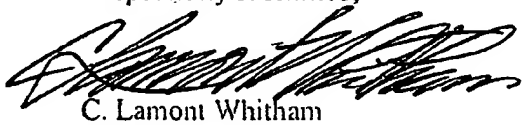
oriented language.

In view of the foregoing, it is respectfully requested that the application be reconsidered, that claims 1 to 24 be allowed, and that the application be passed to issue.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

A provisional petition is hereby made for any extension of time necessary for the continued pendency during the life of this application. Please charge any fees for such provisional petition and any deficiencies in fees and credit any overpayment of fees to Attorney's Deposit Account No. 50-2041.

Respectfully submitted,



C. Lamont Whitham
Reg. No. 22,424

Whitham, Curtis & Christofferson, P.C.
11491 Sunset Hills Road, Suite 340
Reston, VA 20190
Tel. (703) 787-9400
Fax. (703) 787-7557
Customer No.: 30743